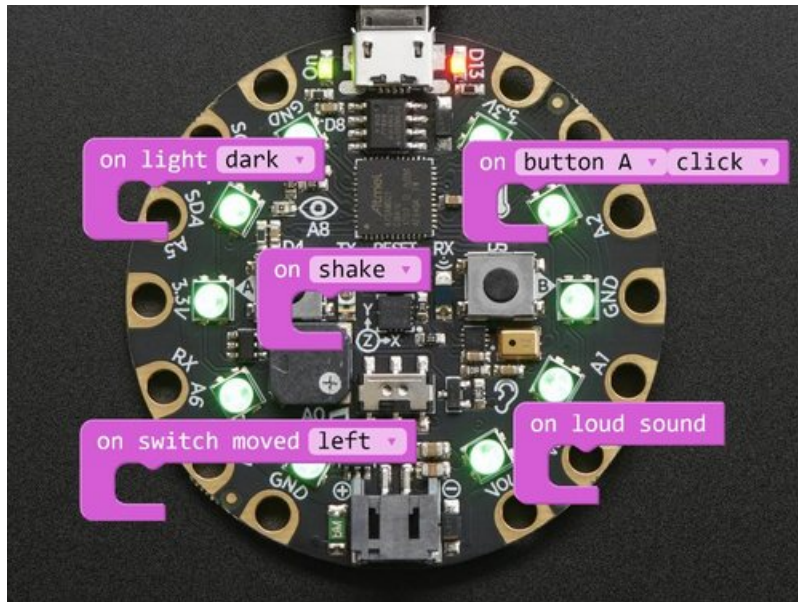


## Sensors in MakeCode

Created by Peli de Halleux



Last updated on 2018-08-22 04:00:55 PM UTC

# Guide Contents

<b>Guide Contents</b>	<b>2</b>
<b>Overview</b>	<b>3</b>
Events vs live data	3
Reference	3
Circuit Playground Express	3
<b>Buttons</b>	<b>4</b>
Events	4
Live data	4
Example	4
<b>Accelerometer</b>	<b>5</b>
Events	5
Live data	5
Example	5
<b>Light sensor</b>	<b>6</b>
Events	6
Live data	6
Example	6
<b>Microphone</b>	<b>7</b>
Events	7
Live Data	7
Example	7
<b>Capacitive Touch</b>	<b>8</b>
Events	8
Live data	8
Example	8
<b>Switch button</b>	<b>9</b>
Event	9
Live data	9
Example	9
<b>Thermometer</b>	<b>10</b>
Events	10
Live data	10
Example	10
<b>Infrared Transceiver</b>	<b>11</b>
Sending data	11
Events	11
Example	11
How does it work?	11

## Overview

The Adafruit Circuit Playground Express comes packed with built-in sensors. This guide will show you how to leverage them in Microsoft MakeCode. If you are new to MakeCode, make sure to [read the MakeCode primer](#).

### Events vs live data

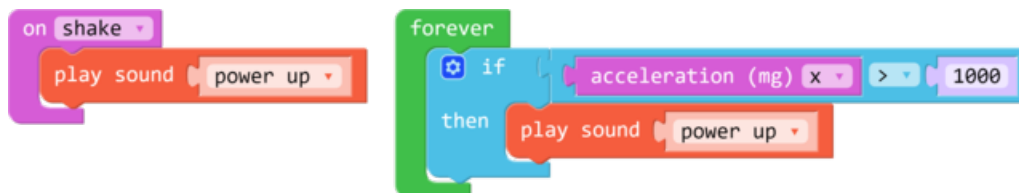
You can use a sensor through **events** or by reading the **live data**.

- **Events** allow to register code that runs when a particular pattern is detected. For example, the **on shake** event runs code when a shacking gesture is detected by the accelerometer. You can mix and match events from different sensors in the same program but each event type can only be registered once.
- **Live data** gets a live (or slightly filtered) reading of the sensor data. For example, **acceleration** gets an immediate reading from the accelerometer.

The block code below show the 2 style of programming with sensors. Through events ( **on shake** left) or in the traditional loop style with live data ( **acceleration** right).

### Reference

Looking for the complete block reference, start at <https://makecode.adafruit.com/reference/input> .



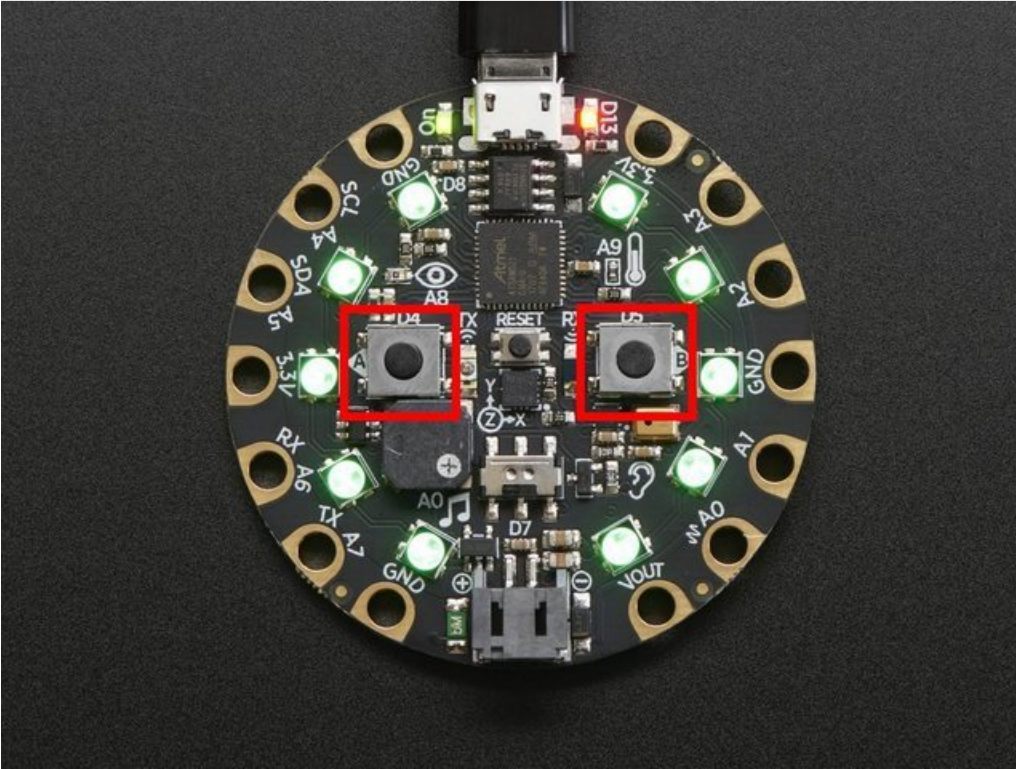
Your browser does not support the video tag.

Circuit Playground Express

\$24.95  
IN STOCK

ADD TO CART

## Buttons



## Events

The `input.onEvent` block allows to respond to clicks, double clicks and other classic button events. It can be mounted on button `A`, `B` or both `A+B` together. Internally, MakeCode takes care of handling the pin state, debouncing, timing and other fun stuff.

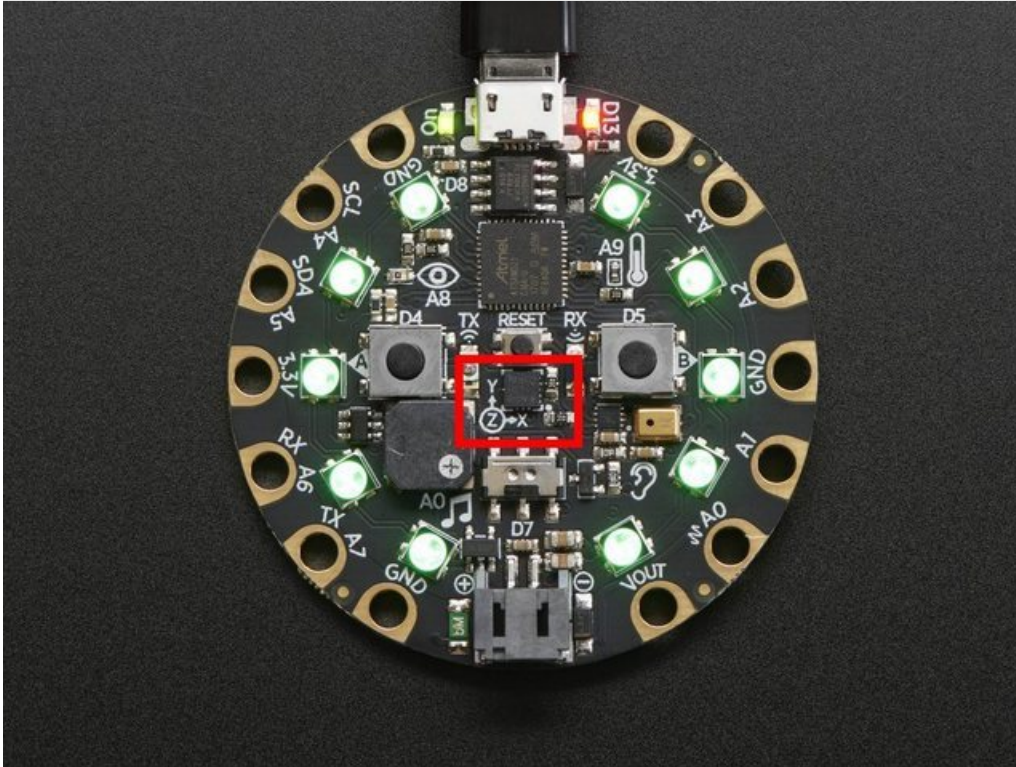
## Live data

You can read the live button status using `input.isPressed`. In some situations, you might miss clicks because your program was busy while the user was pressing. In such case, you can also use `input.wasPressed` which keeps track if the button was pressed between successive calls.

## Example

The `click` event on button `A` is used to `play a sound`. The state of button `B` is checked in a `forever` loop to switch between red and blue on the neopixels.

## Accelerometer



### Events

The `input.onGesture` allows to run code on a number of pre-defined gestures such as **shake**, **freefall** or various orientation events.

### Live data

The `input.acceleration` returns the immediate acceleration for a given direction in **milli-g**, e.g. 1/1000 of a **g**. The measure includes earth gravity (1000mg) You can query **X**, **Y**, **Z** or the **strength**.

If you look closely at the center of the Circuit Playground, you will see the accelerometer axis printed on the board.

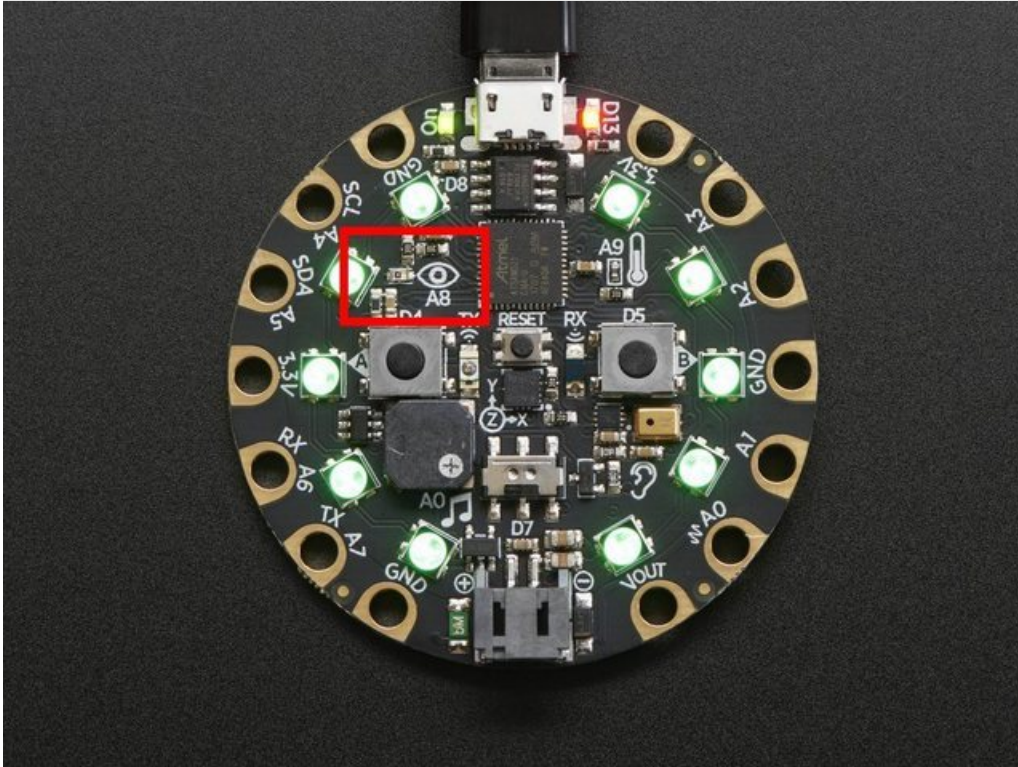
Assuming the board is at rest on a table,

- the **X** axis is aligned horizontally from left to right. If you tilt left, **X** is negative, tilt right **X** is positive.
- the **Y** axis is aligned vertically from bottom to top. If you tilt forward, **Y** is positive, tilt backward **Y** is negative.
- the **Z** axis is perpendicular to the board and pointing down. At rest, **Z** is aligned with earth gravity.

### Example

The example below **plays a sound** when the Circuit Playground is shaken. In a **forever loop**, it display the accelerometer reading using **graph**.

## Light sensor



### Events

The `on light condition changed` event allows to run code when the light goes dark or bright. For example, you can use the `on light bright` event to detect a sudden flash of light.

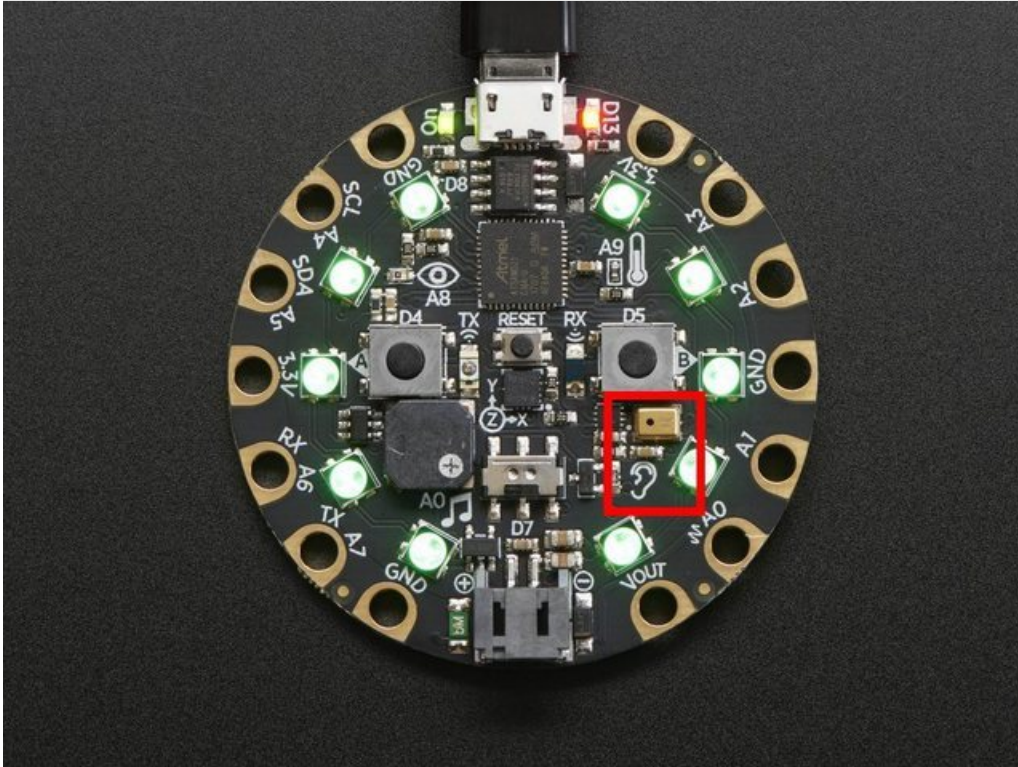
### Live data

The `light level` returns the current light intensity reading between `0` (no light) and `255` (max light).

### Example

In this example, we add a `on light bright` event to run the `sparkle` animation when a flash is detected. Using a `forever` loop, we use the light intensity to control the pitch of the `tones played on the speaker`.

## Microphone



### Events

The `on loud sound` event detects a peak in the sound level, for example when someone claps. You can change the loud threshold using `set loud sound threshold`.

### Live Data

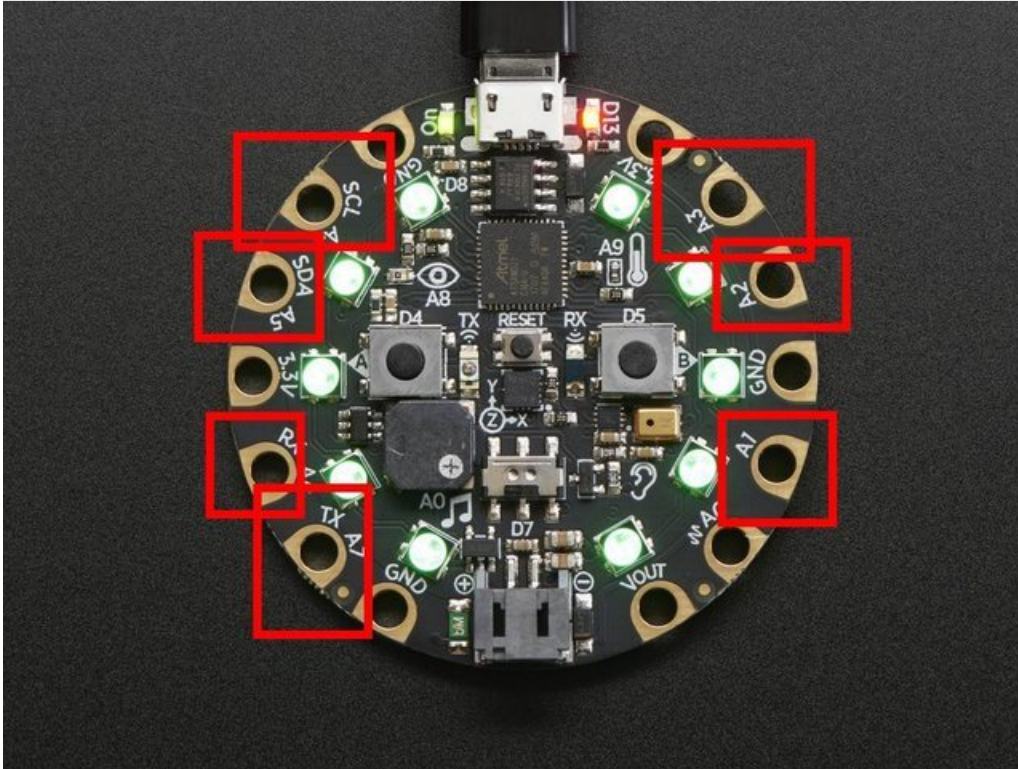
The `sound level` block returns the current sound intensity from `0` (silent) to `255` (very loud).

### Example

The example plays a power up sound when a loud sound is detected. In a `forever` loop, it continuously charts the sound level using the `chart` block.

**WORK AREA** The microphone on the Circuit Playground is capable of sampling sounds. This feature is not yet supported in MakeCode, stay tuned.

## Capacitive Touch



Capacitive pins can be used as buttons similarly to buttons **A** and **B**. You can use pins **A1**, **A2**, **A3**, **A4**, **A5**, **A6** and **A7**.

### Events

The `input.onEvent` block allows to respond to clicks, double clicks and other classic button events. It can be mounted on any of the capacitive pins. You'd typically use croc-clips to create circuits connected to those pins.

### Live data

You can read the live button status using `input.isPressed` or `input.wasPressed`.

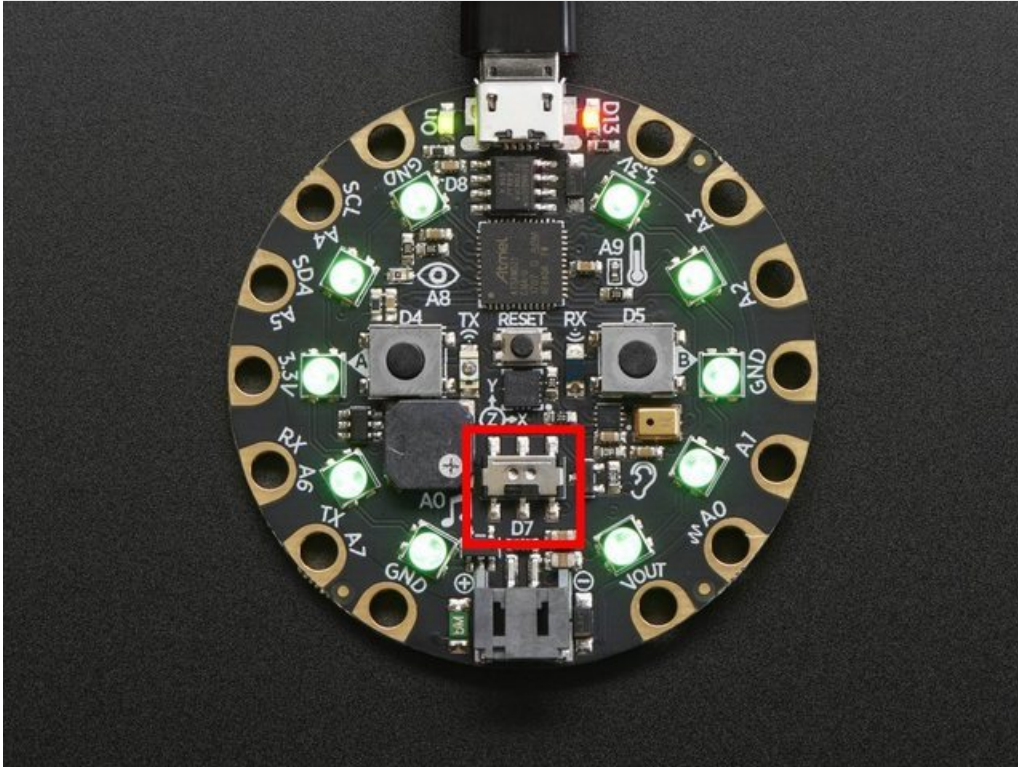
### Example

- the `click` event on pin **A1** is used to `play a sound`
- the state of button **A7** is checked in a `forever` loop to switch between red and blue on the neopixels.

You do not have to ground the pins to get it to work!



## Switch button



The switch button is special kind of button with two positions: left or right. It is very useful to create on/off state in programs.

### Event

The `on switch moved` event runs code when the switch is moved left or right.

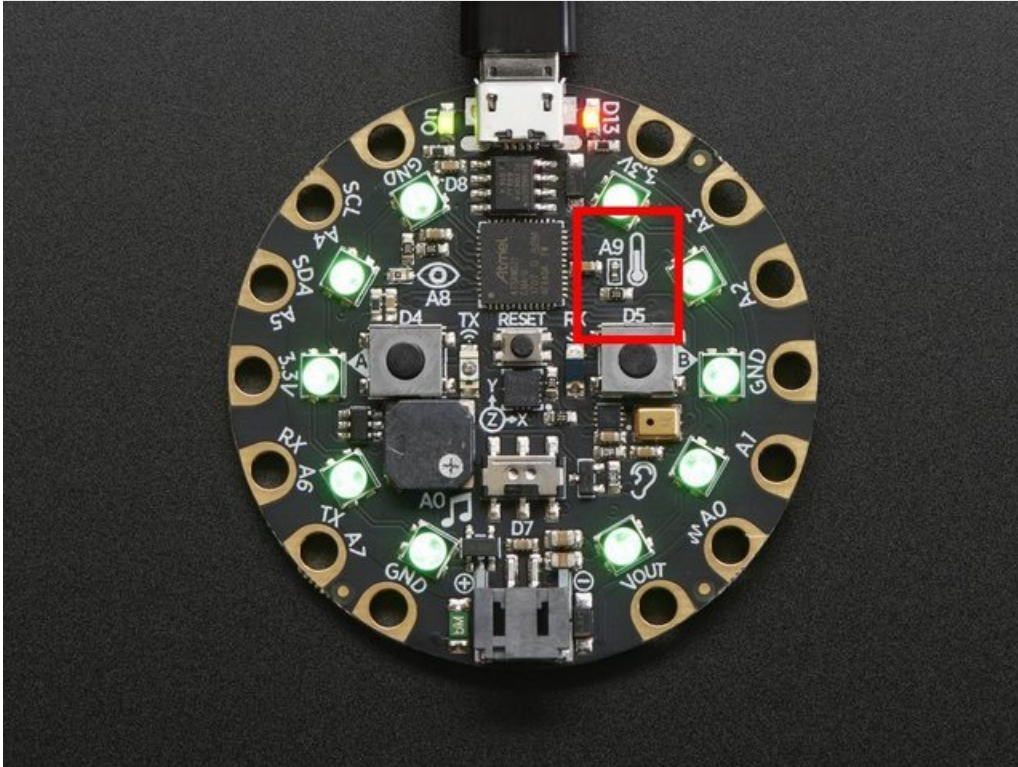
### Live data

The `switch right` block indicates if the switch is on the right position.

### Example

The example below plays different animations when the switch is positioned left or right.

## Thermometer



The onboard thermometer provides an easy way to track the ambient temperature. The MakeCode blocks support Celsius or Fahrenheit degrees.

### Events

The `on temperature hot/cold` event allows to trigger code when hot or cold conditions are detected.

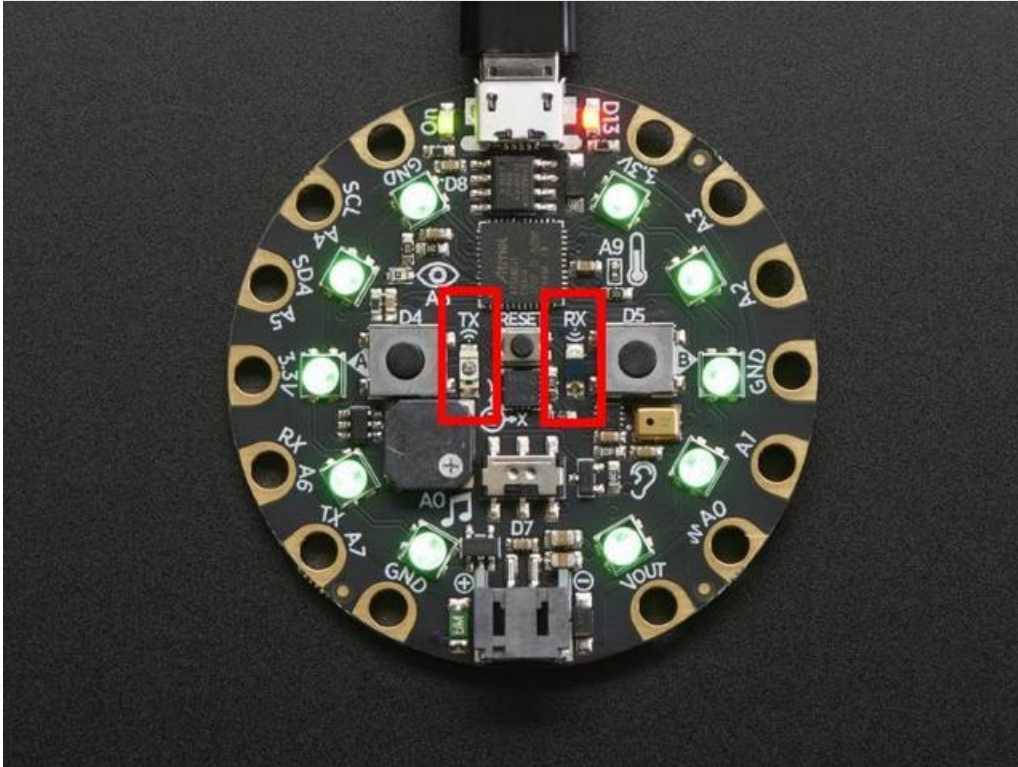
### Live data

The `temperature` block returns the current temperature in Celsius or Fahrenheit degrees.

### Example

The example turns the LEDs to red when the temperature goes above 15. It also graphs the current temperature using the LEDs from 0 to 50C.

## Infrared Transceiver



The onboard Infrared transmitter and receiver diodes allow communicate small chunks of data between Circuit Playgrounds.

### Sending data

The `infrared send number` send a number packet over IR. It may or may not be received.

### Events

The `on infrared received` event triggers when a number has been received.

### Example

The example shows how a rgb color can be sent over IR and used to turn on the neopixel remotely. Button A sends blue, button B sends red.

#### How does it work?

Long story short: it's pretty exciting stuff. [Read the deep dive on the MakeCode blog.](#)